REMARKS

We have amended the claims to more particularly point out and distinctly claim the invention. We have also added claims 32-39 which are dependent claims. After entering these amendments, claims 1-39 will be pending in this application.

The Examiner continues to reject the claims 1-31 under 35 U.S.C. §103(a) as being obvious in view of Carusone (U.S. 5,157,667) over Reynolds (U.S. 6,138,161). We continue to believe, for many of the reasons presented by us before, that the Examiner's rejection is unsupported by the references. In rejecting our claims this time, the Examiner did not explain why our arguments were not correct. So, we are uncertain as to where the disagreement if any might lie. Therefore, we will again try to explain why we believe that Carusone's system does not support a rejection of the claims.

Fundamental Difference Between Carusone and the Claimed Invention

As noted before, we believe that prior to examining what is missing from the Carusone system, it is first important to understand what the fundamental difference is between the approach of the claimed invention as compared to the approach taught by Carusone. To perform fault isolation, Carusone relies on reports sent to a central location by each of the network units that detects the fault. That is, remote devices including neighbors of a failed device send in reports to a central location. In contrast, according to the claimed invention, a computer at the central location identifies failed devices by first <u>attempting to communicate</u> with a target device and if that fails, <u>attempting to communicate</u> with its neighbors. In the embodiment described in the specification attempting to communicate is accomplished by sending a "ping" to the remote device. So, in short, in Carusone's system the network units distributed throughout the network initiate failure reporting; whereas, in the described embodiment, the central location tries to initiate communication with or "pings" the network units.

How Carusone's System Works

In Carusone's system, each of the network units includes a link adapter for each link to which it is connected. Each network unit stores in its local memory what Carusone refers to as "nearest neighbor" information:

> ...each unit of a pair of link coupled units, initially or on reconnection, interrogates a link adapter at the other end of the link for an identifier that identifies both the remote unit and the remote link adapter. This "nearest neighbor" information is stored locally at each unit, and is transmitted to the central location when an error is detected. (Abstract)

Upon detecting a failure, the network unit sends a failure report to the central location:

> ...according to the invention, whenever a failure occurs, failure reports are sent by each unit that observes the failure, to a central location. Each failure report includes the LAID of the link adapter that detected the failure as well as the LAID of the link adapter at the other end of the link (the previously stored LAID of the nearest neighbor). (Col. 5, lines 19-25).

> ...each LAID is basically a unit ID plus a unique number indicating a specific adapter on that unit (Col. 8, line 64 to col. 9, line 2).

In Carusone's system, the central location collects the received failure reports that are sent to it and then analyzes those reports to isolate the source of the problem.

How the Described Embodiment of the Invention Works

In the described embodiment, a host computer 2 periodically "pings" each of the units on the network by sending a packet to that unit. If the target unit does not respond to the ping, thus indicating a failed attempt to communicate, the host computer identifies the neighbors of the target unit. It does this by consulting a network neighbor table which it had previously generated. Then, the host computer pings each of the neighbors, in turn, to determine if any of them is active. If any one of them is active, the host computer concludes that the target unit is a failed unit, i.e., is the root cause of the failed attempt to communicate with it. If none of the neighbors is active, then the host computer concludes that the target unit is not the root cause of the failed attempt to communicate. The host computer then selects one of the inactive neighbors as a new target unit and pings each of its neighbors. This process continues until the host computer finds a target unit that has an active neighbor, at which point the host computer identifies that target unit as the root cause of the failed attempts to communicate. (see pages 8-10 of the specification).

The Differences Between the Claimed Invention and Carusone's System

The Examiner recognizes that, like the claimed invention, Carusone also uses neighbor or nearest neighbor information. The claimed invention, however, does not and is not intended to cover the general concept of using nearest neighbor information to troubleshoot network problems. Rather, it is directed to the more limited concept of using neighbor information to identify targets with which it will attempt to communicate (e.g. by "pinging") them so as to isolate the failed device. Though both systems use neighbor information, they use that information in very different ways. Carusone uses neighbor information to analyze and interpret the meaning of the failure reports that are collected by the central location. The claimed invention uses the neighbor information to identify the targets to ping in order to isolate the root cause of a device failure.

We now turn to the specific differences between what is claimed and what is taught by the two references upon which the Examiner relies. With regard to claim 1, we submit that there is no support whatsoever in Carusone for concluding that if "attempting to communicate" with a target device fails that there is any entity within the Carusone system that then performs the function of "determining if the target device has an active neighbor." Determining if the target device has an active neighbor simply plays no role whatsoever in Carusone's technique for isolating the fault. Furthermore, there is absolutely no support for characterizing the Carusone system as "identifying the target device as a failed device" in response to finding that the target device has an active neighbor.

To recognize that what we have just argued is indeed correct, it is worthwhile to try to map claim 1 onto a hypothetical system that operates in accordance with the teachings of Carusone. For our hypothetical system, assume that there are multiple units on a network, including unit A, unit B, which is a neighbor to unit A, and unit C, which is a computer at a "central location." Also assume that unit A attempts to communicate with neighboring unit B and that attempt fails. According to Carusone, unit A will then send a fault report to entity C. The fault report will at least include the LAID (link adapter ID) for unit A and for unit B, as well as a brief description of the fault that was detected. All of the other entities on the network that "see" the same fault will also report back to unit C in a similar manner. After all of the

generated the fault reports are received, unit C analyzes them and identifies the nature and cause of the fault.

On way of attempting to map the claim onto this hypothetical system might be to view unit B as the "target device." In that case, unit A's attempt to communicate with unit B might correspond to "attempting to communicate with a target device," as required by claim 1. Under this approach, however, there will not be any subsequent operation by any entity that involves "determining **if** the target device has an active neighbor," as is also required by claim 1. Whether the target device has an active neighbor is not an inquiry or a test that any element in the Carusone system makes. Indeed, the Carusone system does not check to see whether any active neighbors even exist. Rather, it simply gathers all failure reports and analyzes them to determine what the fault is and where it is located.

Among the reports the central location receives, there will likely be reports from nearest neighbors of unit B. But simply receiving failure reports sent by active neighbors of unit B certainly is not the same as or equivalent to "determining if the target device has an active neighbor." Furthermore, even if we assumed that it was the same or equivalent, no entity in Carusone's system, upon determining that the target device has an active neighbor, responds by "identifying the target device as a failed device," as is also required by claim 1.

Moreover, unit A's generation of a failure report about the failed communication with unit B means that unit B does have an active neighbor, namely, unit A. If the test of claim 1 were implemented by Carusone (i.e., "if it is determined that the target device has an active neighbor, identifying the target device as a failed device), then every report of a failed communication would result in the system identifying the "target device" as a failed device, which obviously would not be true in Carusone's system.

Another way of attempting to map claim 1 onto the Carusone system might be to view the central location as an active fault report generator. Thus, for example, assume that unit C attempts to communicate with unit B and that attempt fails, thereby potentially satisfying the first element of claim 1. Then, presumably unit C will generate a fault report to itself regarding its failure in attempting to communicate with unit B. But again, even under this scenario, nothing in the Carusone system then attempts to determine whether there is an active neighbor of entity B

and nothing in the Carusone system identifies unit B as a failed device on the basis of finding that it has an active neighbor.

The Examiner also relies on Reynolds to supply that which is missing from Carusone. But nothing in Reynolds supplies the above-identified features that are missing from Carusone.
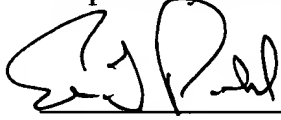
We note that claim 1 is not specific as to what entity is performing each of the individual operations recited in the claim. And indeed, according to the wording of claim 1, the recited functionality could be distributed among multiple entities. Claims 12, 21, and 30, however, do require that the functionality be implemented in a single entity. It should be abundantly clear that Carusone does not teach or even suggest a single entity that has the troubleshooting functionality specified by claims 12, 21 or 30.

For the reasons stated above, we believe that the claims are allowable and therefore ask the Examiner to allow them to issue.

Please charge our Deposit Account No. 08-0219 the required fees of $930.00 for the Petition for Three-Month Extension of Time under 37 CFR 1.136(a), and $144.00 for the eight new dependent claims. Please apply any other charges or credits to Deposit Account No. 08-0219.

Respectfully submitted,

Date: June 2, 2003

Eric L. Prahl
Reg. No. 32,590

Hale and Dorr LLP
60 State Street
Boston, Massachusetts 02109
Telephone: (617) 526-6000
Facsimile: (617) 526-5000